

# Metric Correlation and Analysis Service

## *“Scientific Dashboards”*

*Identify and evaluate areas in monitoring and troubleshooting  
which lack automation and usability features*

*A. Baranovski*

*11/10/2008*

# Index

2

- Concepts and strategy of the project
- Product survey
- Identifiable needs
- Technology survey
- How to move forward

# Concept Areas

3

- Integrating metrics data feeds from various independently owned components
- Selecting or making cuts to multidimensional cube of diagnostic data
- Filtering of system views from general health to details of element status
- User interface. Intuitiveness and ease of use

# Strategy

4

- Component driven software solution
- Adapt, organize access, transformation, and aid display of type agnostic metrics and diagnostic data.
  - Widgetization ( template like UI elements to display details of data )
  - Publishing of data access interfaces
  - Basic data transformation and analysis

# Existing infrastructures

5

- CMS Facilities
- LHC Dashboard
- FermiGrid
- Not nearly a complete list
  - Merely given to distinguish different approaches

# CMS facility

6

- Status display and alarming are built from collection of many independent products
  - Zabbix
    - All in one : DB, display , alarming
  - dCache
    - State plotting tools w/o access to underlying data
  - Condor
    - cmd. tools and wrappers to retrieve and present status information
- Integrated solution supports close to 200000 metrics
- Status is observed by traversing collection of URLs pointing to aspects of interest

# LHC dashboard

7

- What has happened to CMS production job from the perspective of CMS production job
  - Find all jobs with non zero exit code, identify data set used
- Checkpoint information sent to a central database
- Comprehensive GUI front-end
- Behind the scene uses MonaLisa message passing interface and Oracle data store

# FermiGrid

8

- Facility driven product rich in features needed for optimal state reporting and problem detection
  - ▣ Health, summary, and drill down views
  - ▣ Data type agnostic storage of collections of independent metrics
    - File system like navigation of metrics
  - ▣ Integrated workflow engine to poll services for information needed for the display



# Needs

9

- CMS
  - ▣ Dashboard
  - ▣ Troubleshooting
- CMS facility
- RunII

# CMS / Dashboard

10

- Be able to mix together displays of different data sources
  - ▣ Batch queues, job success rates, transfer rates, etc
- Provide facility view configurable by the facility itself
- Health monitoring

# CMS / Troubleshooting

11

- Expand capabilities of working with CMS dashboard
  - ▣ The difficulty with existing interface is inability to mine and present data in a way not predefined by default IU features
    - Challenging to express what you want to look at
- Need a better way to correlate information recorded by the dashboard with other (possibly real time) data sources
  - ▣ May be as simple as having plots and tables conveniently displayed on the same web page

# CMS facility

12

- The operators task involve routine walk through collection of URLs
  - ▣ This is due to use of many independent reporting tools
- Need to aggregate all that into observable health status display.
  - ▣ LeMon at CERN is a good example
- Filtering ( drill down view ) of status information

# RunII

13

- Monitoring is disparate
  - ▣ No common monitoring framework,
  - ▣ No common monitoring host for each experiment.
- Store and reformulate monitoring information to turn it into performance metrics
- Unified mechanism for retaining monitoring so that problems can be examined at some date after the fact.

# Common needs

14

- Synthesis and navigation
- Dashboard Display

# Synthesis and navigation

15

- Use cases where data access needs to span several monitoring realms
  - ▣ Connecting particular job info with storage state
  - ▣ Health function spanning several monitoring providers
  - ▣ Data transformations
- Navigation :
  - ▣ Exploring schema of metrics universe
- \*Hint: Start from published interface to manage and access data

# Dashboard display

16

- A view based on user defined rules to:
  - ▣ Define data
  - ▣ Transform data
  - ▣ Render
  - ▣ Compose final layout out of metrics universe
  - ▣ Create view navigation rules ( or filtering )



# Technology overview

17

- Technology areas most involved in our problem
  - ▣ Data integration
  - ▣ IU Web Widgets

# Data integration questions

18

- Combining data residing at different sources and providing user with a unified view of these data
  - ▣ How to interpret meaning of elements originated from different data sources ?
  - ▣ How to allocate, publish and navigate data schema ?
  - ▣ How to transform or summarize data ?

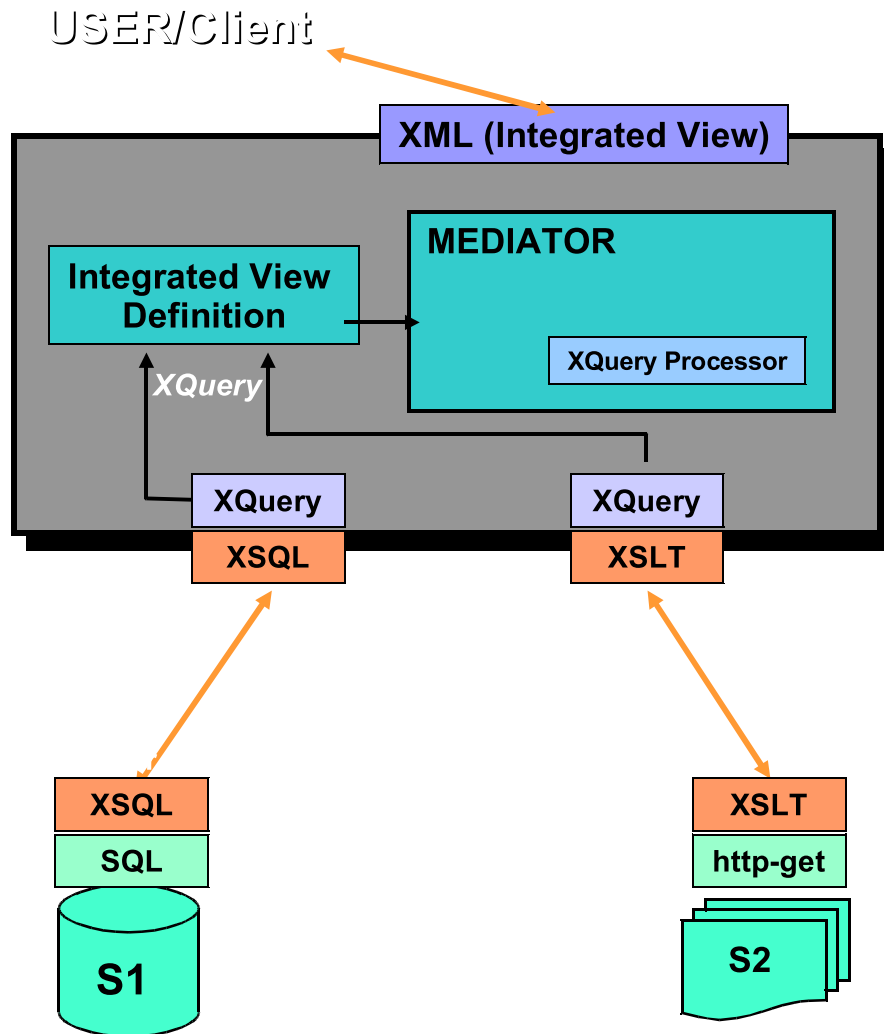
# Approach to Data integration

19

- LAV ( Local as view )
  - ▣ Data sources are viewed through wrapper mediating schema
  - ▣ Queries are dispatched using Mediated Virtual database
    - Example : Integrated environment - Berkley XMLDB
      - XQuery over independent file based collections
- Mediating schema
  - ▣ Best model of the kinds of answers users want from a data source

# XML-Based Mediator System

20



- /root
  - /services
    - /dcache=fapl023....
      - /transferate
      - /total
    - /lsf
      - /queue
- File system like view
  - This layout of the view depends on schema that sets the ontology mapping between observed datasources

# Technology for Dashboard

21

- Desired UI features
  - ▣ Renderer particular metrics using template query
    - May be parametrized by user supplied data
  - ▣ Define hierarchy of filtering rules
  - ▣ Summarize or snapshot existing UI front-ends
- The usual constrain is maximum value with minimum effort
- In this setting , the key requirements to selected technology are capabilities of reuse existing UI and decomposability of the end result

# Technology for Dashboard

22

- “Web Widgets” is a paradigm of development and reuse of self contained UI components which can be organized in accordance to user preferences
  - ▣ A collection of tools and services which help distribution and implementation of coexist-capable and **reusable** Web codes
    - Thus, not a technology or standard in a classical sense
  - ▣ Reliant on existing HTML and Web Script agreements

# Web 2.0 widgets explained

23

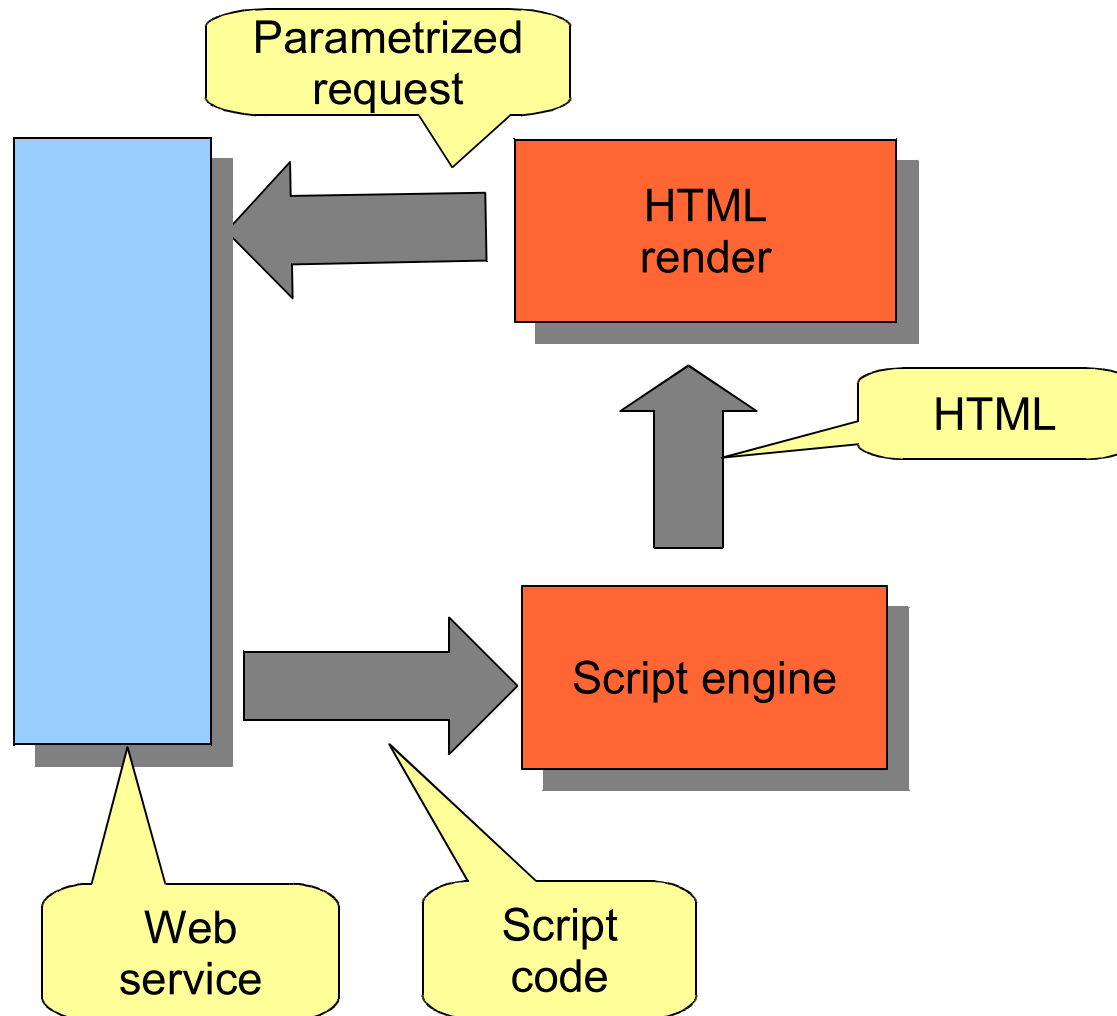
- Embeddable HTML code that requires no additional compilation
- Java Script , Flash or other scripting languages
- The code must be scoped

```
<script  
src="FeedWidget.ashx?  
  URL=<url>/atom.xml"  
  type="text/JavaScript">  
</script>
```



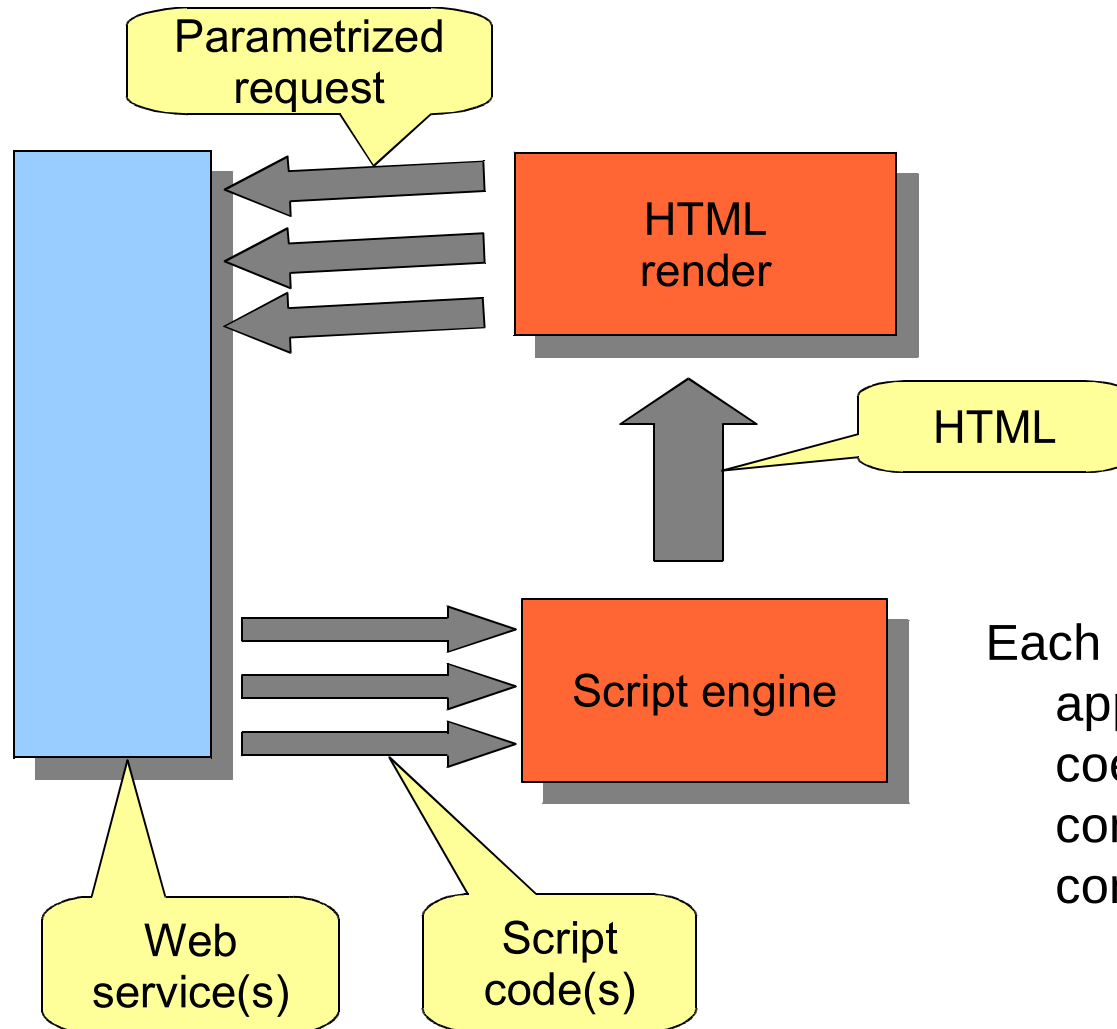
```
document.write('<div class="RssWbox">')  
  
document.write('<p class="RssWtitle"><a class="rss-title"  
  href="" target="_blank">Sidebar Gadgets</a><br  
/></p>')  
  
document.write('<ul class="RssWitems">')  
  
document.write('<li class="RssWitem"><a  
  class="RssWitem" href="introduction-creating-vista-  
gadgets.html" target="_blank">Creating Vista  
Gadgets</a>') document.write('</li>')  
  
document.write('</ul></div>')
```

# Web GUI,traditional architecture



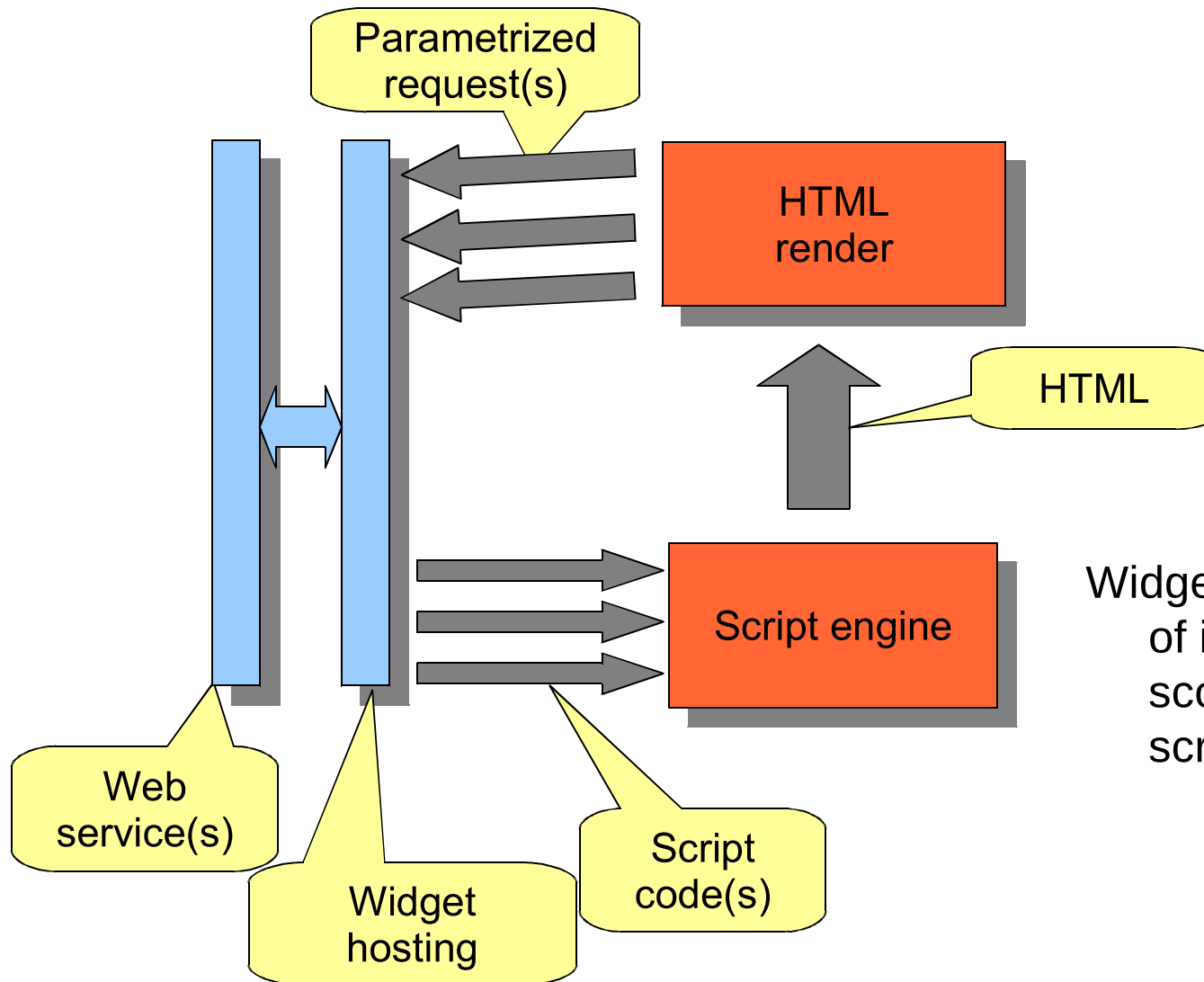


# Widgets based UI



Each code script must be appropriately scoped to coexist with other codes inside common script execution context

# Widgets based UI



Widget hosting is essential level of indirection which ensures scope separation between script codes

# What's next

27

- The project is complicated endeavor with several open ended questions
  - ▣ Primarily in areas of data integration and warehousing
- One tactical approach is to split the project
  - ▣ Phase I
    - Immediate need in composable UI components
    - Health displays (may need some of the features from II)
  - ▣ Phase II and beyond
    - Issues of integrating disjoint monitoring and diagnostic data providers into an abstract portal

# References

28

- Project Web page
  - ▣ <http://www.fnal.gov/docs/products/mcas/>
- Research work on data integration
  - ▣ <http://daks.ucdavis.edu/~ludaesch/Paper/AHM>
- Web Widgets
  - ▣ <http://dev.netvibes.com/doc/>
- Answering queries using views
  - ▣ <http://www.cs.uwaterloo.ca/~david/cs740/ans>